

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method to verify a circuit design, comprising:
~~applying a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and~~
~~determining a behavioral consistency between the first and second computer language representations;~~
~~forming a first bit vector equation in response to unwinding a representation of the circuit design written in a first computer language;~~
~~forming a second bit vector equation in response to unwinding another representation of the circuit design written in a second computer language;~~
~~comparing the bit vector equations; and~~
~~outputting an indication whether the bit vector equations are consistent.~~
2. (Currently Amended) The method of claim 1, ~~wherein applying a bounded model checking technique comprises further comprising:~~
~~jointly unwinding the first and second computer language representations of the circuit design to form a Boolean formula; and~~
~~checking a satisfiability of the Boolean formula~~
~~conjoining the first and second bit vector equations to form a conjoined equation;~~
and
~~passing the conjoined equation to a decision procedure to determine whether the conjoined equation is satisfiable by an assignment of variables within the conjoined equation that makes the equation true.~~

3. (Cancelled)
4. (Currently Amended) The method of claim [[3]] 1, further comprising translating each bit vector equation into a SAT instance.
5. (Original) The method of claim [[2]] 1, further comprising:
forming a Boolean equation from the bit vector equations;
converting the Boolean equation to a conjunctive normal form; and
running a SAT procedure using the conjunctive normal form as an input
~~wherein checking the satisfiability of the Boolean formula comprises applying a SAT procedure.~~
6. (Original) The method of claim 5, wherein comparing the bit vector equations comprises determining a consistency of the bit vector equations.
7. (Original) The method of claim 5, further comprising extracting a counterexample in response to the bit vector equations being inconsistent.
8. (Currently Amended) The method of claim [[2]] 5, further comprising continuing to unwind the ~~computer language~~ representations to find a counterexample in response to the Boolean ~~formula equation~~ not being satisfiable from the SAT procedure.
9. (Original) The method of claim 8, further comprising terminating the method in response to a completeness threshold of a potential counterexample being exceeded.
10. (Currently Amended) The method of claim [[2]] 5, further comprising extracting a counterexample in response to the Boolean ~~formula equation~~ being satisfiable from the SAT procedure.

11. (Original) The method of claim 1, further comprising repeating the method to remove any remaining inconsistencies.

12. (Currently Amended) The method of claim 1, wherein the ~~first computer language representation of the circuit design written in the first computer language~~ comprises a representation of the circuit design written in a hardware description language.

13. (Currently Amended) The method of claim 1, wherein the ~~second computer language representation of the circuit design written in the second computer language~~ comprises a representation of the circuit design written in a C type language.

14. (Currently Amended) A method to verify a circuit design, comprising:
unwinding a ~~first computer language~~ representation of the circuit design written in a first computer language to form a first bit vector equation;
unwinding a ~~second computer language~~ another representation of the circuit design written in a second computer language to ~~from~~ form a second bit vector equation; and
comparing the first and second bit vector equations for consistencies; and
outputting an indication whether the first and second bit vector equations are consistent.

15. (Original) The method of claim 14, further comprising translating each bit equation into a SAT instance.

16. (Original) The method of claim 14, further comprising extracting a counterexample in response to the bit vector equations being inconsistent.

17. (Original) The method of claim 14, further comprising continuing to unwind the computer language representations in response to the bit vector equations being inconsistent.

18. (Original) The method of claim 14, wherein comparing the bit vector equations comprises translating the unwound computer language representations into a Boolean formula.

19. (Currently Amended) The method of claim 18, further comprising determining a satisfaction of the Boolean formula from a SAT procedure.

20. (Cancelled)

21. (Currently Amended) The method of claim 19, further comprising extracting a counterexample in response to the Boolean formula being satisfiable from the SAT procedure.

22. (Currently Amended) The method of claim 19, further comprising continuing to unwind the computer language representations to find a counterexample in response to the Boolean formula not being satisfiable from the SAT procedure.

23. (Original) The method of claim 14, further comprising terminating the method in response to a completeness threshold of a potential counterexample being exceeded.

24. (Original) The method of claim 14, further comprising repeating the method to remove inconsistencies.

25. (Original) The method of claim 14, wherein the first computer language representation is a hardware design language.

26. (Original) The method of claim 14, wherein the second computer language representation is a C type programming language.

27. (Cancelled)
28. (Cancelled)
29. (Cancelled)
30. (Cancelled)
31. (Cancelled)
32. (Cancelled)
33. (Currently Amended) A system to verify a circuit design, comprising:
~~a processor adapted to apply a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and unwind a representation of the circuit design written in a first computer language to form a first bit vector equation, unwind another representation of the circuit design written in a second computer language to form a second bit vector equation, and compare the first and second bit vector equations for consistencies; and~~
~~a software program to determine a behavioral consistency between the first and second computer language representations;~~
~~an output device to output an indication whether the first and second bit vector equations are consistent.~~
34. (Currently Amended) The system of claim 33, wherein the processor is adapted to jointly unwind the first and second computer language representations of the circuit design to form a Boolean formula ~~and to check a satisfiability of the Boolean formula.~~
35. (Original) The system of claim 34, further comprising a SAT solver to check the satisfiability of the Boolean formula.

36. (Currently Amended) The system of claim 33, further comprising:
~~a first bit vector equation formed in response to unwinding the first computer language representation;~~
~~a second bit vector equation formed in response to unwinding the second computer language representation; and~~
means for comparing the bit vector equations for consistency.

37. (Original) The system of claim 33, wherein the first computer language representation of the circuit design comprises a hardware description language.

38. (Original) The system of claim 33, wherein the second computer language representation of the circuit design comprises a C type language.

39. (Currently Amended) A computer-readable medium having computer-executable instructions for performing a method, comprising:
~~applying a bounded model checking technique to a first computer language representation of the circuit design and to a second computer language representation of the circuit design; and~~
~~determining a behavioral consistency between the first and second computer language representations;~~
forming a first bit vector equation in response to unwinding a representation of the circuit design written in a first computer language;
forming a second bit vector equation in response to unwinding another representation of circuit design written in a second computer language;
comparing the bit vector equations; and
outputting an indication whether the bit vector equations are consistent.

40. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 39, further comprising:

jointly unwinding the first and second computer language representations of the circuit design to form a Boolean formula; and

checking a satisfiability of the Boolean formula using a SAT procedure.

41. (Cancelled)

42. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising:

forming a first bit vector equation in response to unwinding the first computer language representation;

forming a second bit vector equation in response to unwinding the second computer language representation; and

comparing the bit vector equations.

43. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 42, wherein comparing the bit vector equations comprises determining a consistency of the bit vector equations.

44. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 43, further comprising extracting a counterexample in response to the bit vector equations being consistent.

45. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising continuing to unwind the computer language representations to find a counterexample in response to the Boolean formula not being satisfiable.

46. (Original) The computer-readable medium having computer-executable instructions for performing the method of claim 40, further comprising extracting a counterexample in response to the Boolean formula being satisfiable.

47. (New) The method of claim 2, further comprising outputting an indication that the circuit design written in the first computer language is consistent with the circuit design written in the second computer language in response to the conjoined equation not being satisfiable.

48. (New) The method of claim 47, further comprising using the circuit design written in the second computer language to simulate and test the circuit in response to conjoined equation not being satisfiable, wherein the first computer language is a hardware description language and the second computer language is a programming language.